PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/00 | A2

(11) International Publication Number: WO 99/14893

(43) International Publication Date: 25 March 1999 (25.03.99)

(21) International Application Number: PCT/US98/19169

(22) International Filing Date: 15 September 1998 (15.09.98)

(30) Priority Data:

60/059,171	17 September 1997 (17.09.97)	US
08/947,081	8 October 1997 (08.10.97)	US
09/050,750	30 March 1998 (30.03.98)	US

(71) Applicant *(for all designated States except US)*: SONY ELECTRONICS INC. [US/US]; 1 Sony Drive, Park Ridge, NJ 07656 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: CHUNG, David, H. [US/US]; Apartment 1901, 243 Buena Vista Avenue, Sunnyvale, CA 94086 (US).

(74) Agents: HAVERSTOCK, Thomas, B. et al.; Haverstock & Owens LLP, Suite 420, 260 Sheridan Avenue, Palo Alto, CA 94306 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

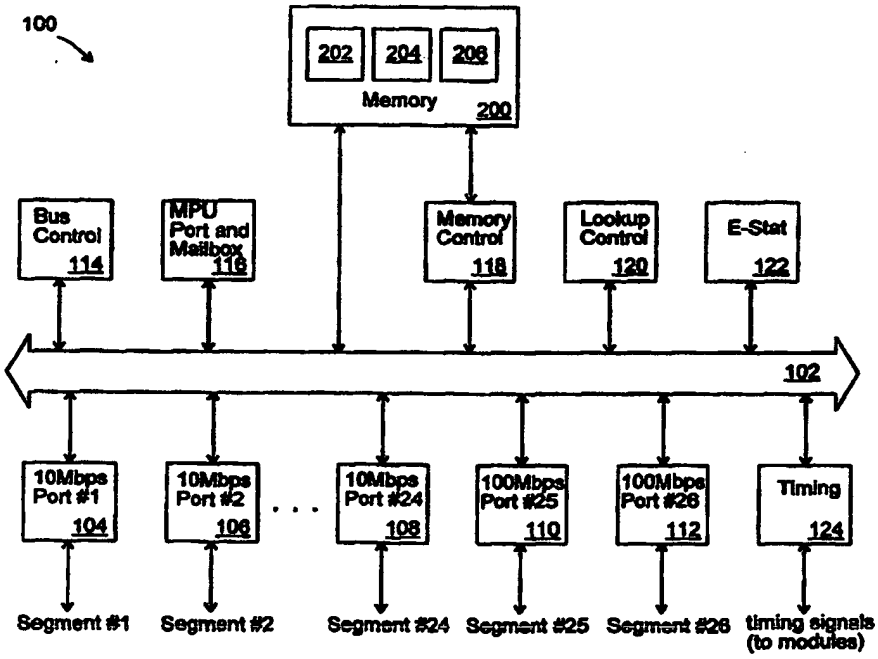
Published

Without international search report and to be republished upon receipt of that report.

(54) Title: TRIPLET ARCHITECTURE IN A MULTI-PORT BRIDGE FOR A LOCAL AREA NETWORK INCLUDING METHOD AND APPARATUS FOR PERIODICALLY UPDATING ENTRIES IN A CONTENT ADDRESSABLE MEMORY

(57) Abstract

A multi-port bridge includes a memory and a plurality of ports. Each port includes a receive buffer, a transmit buffer and a "triplet" buffer. As a data packet is being received by the receive buffer of a port, a look-up table is utilized to identify the appropriate destination port for the packet. A result of the look-up is a "triplet" which includes: a first field containing the identification of the source port, a second field containing the identification of the destination port, and a third field containing a starting address assigned to the incoming packet in the memory. The triplet is placed upon the communication bus a first time. If the destination port is available, the destination port receives the packet simultaneously as the packet is stored in the memory. Otherwise, the triplet is placed on the communication bus a second time after the packet is stored in the memory. The destination port stores the triplet in its triplet buffer. Then, when the destination port is available, the destination port retrieves the packet from the memory for transmission.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

TRIPLET ARCHITECTURE IN A MULTI-PORT BRIDGE FOR A LOCAL AREA NETWORK INCLUDING METHOD AND APPARATUS FOR PERIODICALLY UPDATING ENTRIES IN A CONTENT ADDRESSABLE MEMORY

This is a continuation-in-part of U.S. Patent Application Serial No. 08/590,125 filed on January 23, 1996, which is a continuation-in-part of U.S. Patent Application Serial No. 08/371,499, filed on January 11, 1995. The contents of U.S. Patent Application Serial No. 08/590,125 and the contents of U.S. Patent Application Serial No. 08/371,499 are hereby incorporated by reference. This application claims the benefit of U.S. Provisional Application No. 60/059,171, filed September 17, 1997.

Field of The Invention:

The invention relates to a multi-port bridge for a local area network. More particularly, the invention relates to controlling flow of data packets among the ports of a multi-port bridge. The invention also relates to the field of look-up tables for appropriately directing data packets through a multi-port bridge in a local area network. More particularly, the invention relates to the field of periodically updating entries in a look-up table for appropriately directing data packets through a multi-port bridge in a local area network where the look-up table is a random access memory (RAM) device which is configured to emulate a content addressable memory (CAM) device.

Background of the Invention:

Nodes of a local area network (LAN) are typically interconnected by a shared transmission medium. The amount of data traffic that the shared transmission medium can accommodate, however, is limited. For example, only one node at a time can successfully transmit data to another node over the shared transmission medium. If two or more nodes simultaneously attempt to transmit data, a data collision occurs, which tends to corrupt the data being transmitted. Thus, nodes that share a transmission medium are considered to be in a same collision domain.

A multi-port bridge allows simultaneous communication between nodes of the LAN by segmenting the LAN into multiple collision domains (also referred to as network segments), each segment having a corresponding transmission medium.

Fig. 1 illustrates a conventional local area network including a multi-port bridge 10. The multi-port bridge 10 has eight ports A-H, though the number of ports can vary. Each port A-H is connected to a segment 11-18 of the LAN. Each segment 11-18 typically includes one or more nodes 19-34, such as a workstation, a personal computer, a data terminal, a file server, a printer, a facsimile, a scanner or other conventional digital device. Each of the nodes 19-34 has an associated node address which uniquely identifies the node. The nodes 19-34 are configured to send data, one to another, in the form of discrete data packets.

When the LAN operates according to Ethernet standards, such as the Institute of Electrical and Electronics Engineers (IEEE) 802.3 standard, data is communicated in the form of discrete packets. Fig. 2 illustrates a conventional IEEE 802.3 data packet 40. The data packet 40 includes an eight byte long pre-amble 41 which is generally utilized for synchronizing a receiver to the data packet 40. The pre-amble 41 includes seven bytes of pre-amble and one byte of start-of-frame. Following the pre-amble 41, the data packet 40 includes a six-byte-long destination address 42, which is the node address of a node which is an intended recipient for the data packet 40. Next, the data packet 40 includes a six-byte-long source address 43, which is the node address of a node which originated the data packet 40. Following the source address 43 is a two-byte length field 44. Following the length field 44 is a data field 45. The data field 45 can be up to 1500 bytes long. Finally, the data packet 40 includes a two-byte frame check field 46 which allows a recipient of the data packet 40 to determine whether an error has occurred during transmission of the data packet 40.

When a node (source node) sends data to another node (destination node) located on its same segment of the LAN (intra-segment communication), the data is communicated directly between the nodes without intervention by the multi-port bridge 10 and is known as an intra-segment packet. Therefore, when the multi-port bridge 10 receives an intra-segment packet, the multi-port bridge 10 does not bridge the packet (the packet is filtered). When a node (source node) sends a data packet to another node (destination node) located on a different segment (inter-segment communication), the multi-port bridge 10 appropriately forwards the data packet to the destination node.

Data packets received by the multi-port bridge 10 are generally stored temporarily within the multi-port bridge 10 before being forwarded to the appropriate destination node

by the multi-port bridge 10. Problems can arise, however, when the capabilities of the multi-port bridge are exceeded by network demand. When data packets are received by the multi-port bridge 10 at a rate that is higher than the rate at which the multi-port bridge 10 can appropriately forward the packets, the network becomes congested. This problem is exacerbated as network users place increasing demands on the network.

Therefore, what is needed is improved technique for controlling the flow of data in a multi-port bridge for a local area network.

Random access memory (RAM) devices generally comprise a plurality of data storage locations, each data storage location having a corresponding predetermined address. Fig. 3 illustrates a conventional RAM device 50. As illustrated in Fig. 3, the RAM device 50 has address fields that are each eight bits in length. Accordingly, the RAM device 50 has 256 unique addresses. Because there is a one-to-one correspondence of addresses to data storage locations, there are 256 data storage locations (fields) in the RAM device 50. The RAM device 50, however, could have other address field lengths and, thus, a different number of unique addresses and corresponding storage locations.

The RAM device 50 includes an address port 52, a data port 54 and a read/write control input 56. To store data in a data storage location of the RAM device 50, the address of the location in which the data is to be stored is applied to the address port 52 and the data to be stored is applied to the data port 54. Then, the read/write control input 56 is conditioned to write the data into the location specified by the address. To read data from the RAM device 50, the read/write control input 56 is conditioned to read data and the address of the location from which the data is to be read from is applied to the address port 52. The data stored in the location specified by the address appears at the data port 54.

Content addressable memory (CAM) devices generally comprise a plurality of data storage locations, each data storage location having a predetermined corresponding address. CAM devices are characterized in that each data storage location includes a comparand field and an associated data field. Comparands are stored in the comparand fields, while data associated with each comparand is stored in the associated data field. Data from the associated data fields is read by comparing each comparand to an applied value. When a match occurs, the contents of the associated data field corresponding to the matching comparand are read.

Fig. 4 illustrates a conventional CAM device 60. As illustrated in Fig. 4, the CAM device 60 has address fields that are each eight bits in length. Accordingly, the CAM device 60 has 256 unique addresses. Because there is a one-to-one correspondence of addresses to data storage locations, there are 256 data storage locations in the CAM device 60, each data storage location including a comparand field and an associated data field. The CAM device 60, however, could have other address field lengths and, thus, a different number of unique addresses and corresponding storage locations.

The CAM device 60 includes an address port 62, a comparand port 64, an associated data port 66, a read/write control input 68 and a match output 69. To store a comparand and associated data in the CAM device 60, a desired address is applied to the address port 62, the desired comparand is applied to the comparand port 64 and the desired associated data is applied to the associated data port 66. Then, the control input 68 is conditioned to write the comparand and the associated data into the location specified by the address.

To read associated data from the CAM device 60, the control input is conditioned to read data and a specified value is applied to the comparand port 64. If a stored comparand matches the specified value applied to the comparand port 64, this condition will be indicated by the match output 69. Data stored in the associated data field which corresponds to the matching comparand appears at the associated data port 66. Otherwise, if no comparand matches the value applied to the comparand port 64, the match output 69 will indicate this condition.

Therefore, to read associated data from the CAM device 60, the value applied to the comparand port 64 is compared to the previously stored contents of all the comparand fields. When a match is found, the associated data appears at the data port 66. Accordingly, the memory device 60 is referred to as "content addressable."

Because of this "content addressable" feature, CAM devices are typically more expensive to manufacture and to purchase than RAM devices, however, CAM devices are particularly well suited for certain applications. For example, a CAM device can be utilized to construct a look-up table for appropriately directing data packets through a multi-port bridge in a local area network (LAN).

In particular, the multi-port bridge 10 (Fig. 1) receives each data packet 40 (Fig. 2) and must determine whether the data packet 40 is for intra-segment communication or

inter-segment communication, and if the data packet 40 is for inter-segment communication, the multi-port bridge 10 must determine which port (destination port) the data packet 40 is to be directed based upon the destination address 42 contained in the data packet 40. This can be accomplished with a look-up table constructed in a CAM device 60 (Fig. 4). Conventionally, the look-up table is constructed by executing a learning phase for each received data packet 40 to store data in the table and by executing a look-up phase for each received data packet 40 to look-up data stored during the learning phase for a prior packet.

An example is provided to enhance understanding of the learning phase. To understand the example, one should refer to Figs. 1, 2 and 4. The learning phase is executed by storing in the address field of the CAM device 60 the address of the node (source node) which sends a data packet 40 and storing in the corresponding data field the number of the port (source port) which received the data packet 40. For example, referring to Fig. 1, if the node 19 sends a data packet 40 to the node 30, the data packet 40 will include the address of the node 19 as its source address. The multi-port bridge 10 will receive the data packet 40 from port A. Therefore, for the learning phase, the address of the node 19 will be stored in the address field of the CAM device 60 and a number identifying the port A will be stored in the corresponding data field. This process is repeated for each data packet 40 received by the multi-port bridge 10.

Once the look-up table is constructed, to appropriately direct a next data packet 40, the destination address contained in the data packet 40 is applied to the address port 64 of the CAM, while the data port 66 of the CAM will indicate to which port (destination port) the packet is to be directed. In the example, the node 30 is the intended recipient. Thus, the data packet will include the address of the node 30 as its destination address The node 30 is connected to the port F. Therefore, when the address of the node 30 is applied to the address port 64 of the CAM device 60, the data port 66 of the CAM device 60 will indicate that the data packet 40 should be routed to the port F. Data packets 40 received before the table is completed are broadcast to all the ports A-H to ensure that the appropriate destination node receives the packet.

To save costs, the CAM device 60 can be emulated by a RAM device through a technique known as hashing. Hashing is a mathematical technique of mapping a larger set of numbers into a smaller set. For example, in an Ethernet LAN, each node is specified by

an address that is six bytes (48 bits) long. Therefore, there are 2^{48} (approximately 2.8 x 10^{14}) possible addresses. Using a RAM device to store a port number associated with each of these possible addresses would require a RAM device having 2^{48} memory locations. Construction of such a large RAM device would tend to be impractical and would almost certainly exceed the cost and space requirements of an appropriately sized CAM device. Because Ethernet LANs generally have far fewer than 2^{48} nodes, hashing can be utilized to construct a look-up table having far fewer than 2^{48} entries.

According to the IEEE 802.3 standard, a look-up table is required to accurately reflect nodes which have been newly added or removed from the LAN. For this purpose, each node of the LAN periodically transmits a "keep alive" packet, preferably once every 2.5 minutes. Entries in a look-up table which have not been updated within a predetermined time period, preferably five minutes, are removed from the look-up table.

A known technique for updating such a look-up table in an Ethernet LAN includes storing a time stamp in the look-up table along with every entry and updating the time stamp whenever the corresponding node originates a data packet. In addition, the technique requires keeping track of the age of every entry in the look-up table, and when an entry becomes more than five minutes old, removing the entry. This technique has a disadvantage in that a large amount of processing overhead is required to provide a time stamp for each entry in the table, to update the time stamps, to keep track of the time elapsed since a last update for each entry, and to remove entries after five minutes have elapsed since the last update for the entry. Also, additional storage space is required for storing the time stamps.

Therefore, what is needed is an improved technique for periodically updating the entries in a CAM device. What is further needed is an improved technique for periodically updating the entries in a CAM device emulated by a RAM device. What is still further needed in an improved technique for periodically updating the entries in a look-up table utilized for appropriately directing data through a multi-port bridge in an Ethernet LAN.

Summary of the Invention:

The invention is a method and apparatus for controlling flow of data packets among the ports of a multi-port bridge. The multi-port bridge includes a switch engine, a dynamic random access memory and a plurality of ports, all of which are interconnected by a high

speed communication bus. The switch engine includes a memory controller and a bus controller, each being a finite state machine. The bus controller controls access to the communication bus by collecting requests from the ports and granting the requests according to an appropriate priority. The memory controller provides an interface between the memory and the communication bus. The memory includes look-up tables utilized for appropriately directing data packets among the ports, packet buffers utilized for temporarily storing packets and mailboxes for providing an interface between the switch engine and an external processor.

Each port includes a port controller, a MAC transceiver, a receive buffer, a transmit buffer and a triplet buffer. Packets received from a LAN segment by the transceiver are directed to the communication bus through the receive buffer, while packets to be transmitted over the LAN segment are directed to the transceiver through the transmit buffer. The triplet buffer stores memory pointers, referred to herein as "triplets," for data packets being queued in the packet buffers of the memory.

A data packet originating from a node (source node) in a segment of the LAN is received by the receive buffer of a corresponding one of the ports (source port) of the multi-port bridge. As the packet is still being received, the look-up tables are utilized to determine which is the appropriate destination port for the packet based upon the destination address. A result of performing the look-up is a "triplet" which includes three fields: a first field containing the identification of the source port, a second field containing the identification of the destination port, and a third field containing a starting address assigned to the incoming packet in the packet buffers of the memory.

If the source port and the destination port are the same, this indicates that the source and destination nodes are on the same segment of the LAN (intra-segment communication) and the packet is filtered. Otherwise, the memory controller places the triplet on the communication bus which is monitored by each port. If the port identified as the destination port in the triplet is not currently busy transmitting or receiving another packet, the destination port will configure itself to receive the packet directly from the source port (cut-through).

However, if the triplet buffer in the port identified as the destination port in the triplet is nearly full, the bus controller of the destination port applies a jam request signal to the communication bus. The source port will receive the jam request and, in response,

will discard the incoming packet and also send a jam signal over its associated segment. The jam signal will cause the node (source node) which is the source of the packet to discontinue sending the packet and attempt to resend the packet after a waiting period.

Once the triplet has been formed, the packet is loaded from the receive buffer of the source port into the packet buffers in the memory starting at the memory address identified by the third field of the triplet. Writing of the packet into the packet buffers preferably occurs as the remainder of the packet is still being received into the receive buffer of the source port. For this reason, the receive buffer for each port need not be capable of storing the entire data packet. In addition, if the destination port is configured for cut-through, the destination port will receive the packet into its transmit buffer directly from the communication bus simultaneously with the write cycles for loading of the packet into the packet buffers. During such a cut-through operation, the packet is received into a transmit buffer of the destination port for immediate transmission to the LAN segment associated with the destination port.

Once the entire packet has been loaded into the packet buffers, the triplet is again placed on the data lines of the communication bus. The destination port then stores the triplet in its triplet buffer. Thus, the packet is queued for transmission by the destination port. Then, when the destination port is no longer busy, the destination port retrieves the packet from the packet buffers.

While the destination port is receiving the packet into its transmit buffer from the packet buffers or directly from the source port, the destination port begins transmitting the packet to the LAN segment associated with the destination port. For this reason, the transmit buffer for each port need not be capable of storing an entire data packet The packet is then received from the LAN segment by the destination node for the packet.

A limited number of bus commands are utilized for controlling the flow of packets through the multi-port bridge. In comparison to prior systems, the throughput capacity of the multi-port bridge is improved.

The invention includes a method of and apparatus for periodically updating entries in a look-up table utilized for appropriately directing data packets through a multi-port bridge in a local area network (LAN) and for ensuring that no entry older than predetermined amount of time is utilized to direct a data packet through the multi-port bridge. Each of the data packets is originated by a respective one of a plurality of nodes

of the LAN and contains a source address, which is the address of the node which originated the data packet, and a destination address, which is the address of the node which is intended to receive the data packet. One or more nodes are included in each segment of the LAN and each segment is coupled to a corresponding port of the multi-port bridge. Each node can be a workstation, a personal computer, a data terminal, a file server, a printer, a scanner, a facsimile or other conventional digital device. Each node is identified by a unique node address, while each port is identified by a unique port number.

When a node (source node) in the LAN sends a data packet to another node (destination node) in the LAN, the data packet is received by a port (source port) of the multi-port bridge. A look-up table is used for identifying the location of the destination node. The look-up table is constructed by performing a learning phase for each data packet in which the number of the port (source port) which receives the data packet is stored in association with the source address contained in the data packet. The multi-port bridge performs a look-up cycle upon the destination address contained in the packet to identify the appropriate other port (destination port) for transmitting the data packet to the node which is to receive the data packet. Then, the multi-port bridge directs the data packet to the port indicated (the destination port).

The look-up table comprises a first memory store, a second memory store, and a third memory store. Each memory store is either a CAM device or a RAM device which emulates a CAM device. A learning phase is performed for each data packet received during a first period of time. During the learning phase, the port identification number of the source port is stored in association with the address of the source node in both the first memory store and in the second memory store. This allows rapid correlation of node to port address information. Also, in a look-up cycle for each data packet received during the first period of time, entries in the first memory store are utilized for directing the data packet through the multi-port bridge (determining which is the appropriate destination port for the packet). At the end of the first period of time, the third memory store is cleared of all entries.

In a learning phase for each data packet received during a second period of time, the port identification of the source port is stored in association with the address of the source node in both the second memory store and in the third memory store. Also, in a look-up cycle for each data packet received during the second period of time, the entries in

the second memory store are utilized for directing the data packet through the multi-port bridge. At the end of the second period of time, the first memory store is cleared of all entries.

In a learning phase for each data packet received during a third period of time, the port identification number of the source port is stored in association with the address of the source node in both the first memory store and in the third memory store. Also, in a look-up cycle for each data packet received during the third period of time, entries in the third memory store are utilized for directing the data packet through the multi-port bridge. At the end of the third period of time, the second memory store is cleared of all entries.

This sequence is repeated, beginning with the first period of time, so long as the multi-port bridge is directing packets. Accordingly, no entry that is more than two time periods old is utilized for directing a data packet through the multi-port bridge. Preferably, the first, second and third time periods are each 2.5 minutes. Therefore, no entry that is more than 5 minutes old is utilized for directing a data packet through the multi-port bridge. Unlike prior systems, the age of each entry in the look-up table need not be individually tracked.

Brief Description of the Drawings:

Fig. 1 illustrates a conventional local area network (LAN) including a multi-port bridge 10.

Fig. 2 illustrates a conventional IEEE 802.3 data packet.

Fig. 3 illustrates a conventional random access memory (RAM) device.

Fig. 4 illustrates a conventional content addressable memory (CAM) device.

Fig. 5 illustrates a block schematic diagram of a multi-port bridge according to the present invention.

Fig. 6 illustrates a port of the multi-port bridge according to the present invention.

Fig. 7 illustrates a "triplet" according to the present invention, including a first field containing an identification of a source port, a second field containing an identification of a destination port and a third field containing a memory address.

Fig. 8 illustrates a block schematic diagram of the switch engine, a memory device and an external processor according to the present invention.

Fig. 9 illustrates a block schematic diagram of an apparatus according to the present invention for periodically updating a look-up table having three look-up sub-tables.

Fig. 10 illustrates a time chart according to the present invention for periodically updating the three look-up sub-tables illustrated in Fig. 9.

Detailed Description of a Preferred Embodiment:

The following documents are hereby incorporated by reference: U.S. Patent Application Serial No. 08/590,125, filed January 23, 1996; U.S. Patent Application Serial No. 08/371,499, filed January 11, 1995; U.S. Patent Application Serial No. 08/947,081, filed October 8, 1997; U.S. Patent Application Serial No. 08/946,866, filed October 8, 1997; U.S. Patent Application Serial No. 09/025,356, filed February 18, 1998; and U.S. Patent Application Serial No. 09/025,355, filed February 18, 1998.

In the preferred embodiment, the present invention is utilized for appropriately directing data packets through a multi-port bridge for an Ethernet LAN. It will be apparent, however, that other devices in an Ethernet LAN, such as a switch or a router, or devices in a network operating according to another networking standard, can utilize the advantages of the present invention.

Fig. 5 illustrates a block schematic diagram of a multi-port bridge 100 according to the present invention. A high speed communication bus 102 provides an interconnection for each of the functional blocks 104-124 of the multi-port bridge 100. The communication bus 102 preferably includes five command lines and thirty-two data lines, though it will be apparent that other bus configurations can be utilized. According to the preferred embodiment, twenty-four 10 Mbps ports 104-108 and two 100 Mbps ports 110-112 are each coupled to the communication bus 102 and can be coupled to a respective LAN segment, each LAN segment having one or more nodes. Each of the twenty-four 10 Mbps ports 104-108 transmit and receive data packets at a rate of 10 Mbps, whereas, the two 100 Mbps ports 110-112 transmit and receive data packets at a rate of 100 Mbps. It will be apparent, however, that other numbers of ports, other port configurations and other performance characteristics can be utilized.

A bus control module 114 controls access to the communication bus 102 by collecting requests from the ports 104-112 and from the other modules. Based upon the requests, the bus control module 114 grants access to the communication bus 102 according

to an appropriate priority. The bus control module 114 also controls access to a memory device 200 by an external processor (MPU), as explained in more detail herein. An MPU port and mailbox module 116 provides an interface between the multi-port bridge 100 and the external processor 500 (Fig. 8) for performing various functions, as will also be explained in more detail herein. These functions include loading data into registers of the multi-port bridge 100, fetching data from registers of the multi-port bridge 100 and transferring data packets between the external processor 500 and the ports 104-112 of the multi-port bridge 100.

A memory control module 118 provides an interface between the memory device 200 and the communication bus 102 and also provides an interface between the memory device 200 and a look-up control module 120. The memory device 200 includes mailboxes 202 for exchanging information between the external processor and the multi-port bridge 100. In addition, the memory device includes look-up tables 204. The look-up tables 204 include entries which indicate which port of the multi-port bridge 100 is associated with each node of the LAN. The look-up tables 204 are utilized for appropriately directing among the ports 104-112 data packets received by the multi-port bridge 100, as will be explained in more detail herein.

The look-up control module 120 receives addresses of nodes and associated port identifications from the communication bus 102. These addresses and identifications are stored in the look-up tables 204. The look-up control module 120 facilitates utilizing the look-up tables 204 for directing packets among the ports 104-112 based upon the destination address of each packet. The memory device 200 also includes packet buffers 206 for temporarily storing data packets that are being directed through the multi-port bridge 100. The memory device 200 is preferably an SDRAM device, though other types of memory devices can be utilized, such as DRAM, SRAM, RAM or EDO. In the case of dynamic memory, the memory control module 118 refreshes the memory device 200 as required.

An E-stat module 122 collects data packet routing statistics and provides them to the external processor for performing analysis and network management functions. A timing module 124 provides timing signals to the ports 104-112 and to the other modules 114-122 of the multi-port bridge 100. Preferably, a primary clock signal cycles at 40

MHz. Other clock signals, at 10 MHz and 25 MHz, are derived from the primary clock signal.

Preferably, the modules 114-124 are each implemented as a finite state machine, though the modules 114-124 can alternately be implemented as one or more processors operating according to stored software programs. Finite state machines are preferred as they can generally perform the necessary operations faster, thus, resulting in a higher packet handling capacity for the multi-port bridge 100.

Fig. 6 illustrates a block schematic diagram of one of the ports 104-112 of the multi-port bridge 100. A port controller 300, including a bus controller 302 and registers 304, provides control for the port and an interface between the port and the communication bus 102. The port controller 300 monitors the communication bus 102 for commands and data directed to the port and also provides commands to the communication bus 102 at times when the port has control of the communication bus 102. The registers 304 contain data for initializing the port upon start-up and for collecting status information for the port. The port also includes a triplet FIFO buffer 306 coupled between the communication bus 102 and the port controller 300. The triplet buffer 306 stores memory pointers ("triplets" - explained in more detail herein) for data packets being queued in the packet buffers 206 (Fig. 5) of the memory device 200. Preferably, the triplet buffer 306 holds 128 triplets, each triplet preferably being four bytes long.

The port also includes a medium access control (MAC) transceiver 308 which accesses a LAN segment 310 for transmitting and receiving data packets to and from the LAN segment 310. Associated with and coupled to the transceiver 308 are a receive finite state machine 312, for controlling the transceiver 308 during packet reception, and a transmit finite state machine 314, for controlling the transceiver 308 during packet transmission. The receive finite state machine 312 and the transmit finite state machine 314 are each coupled to the bus control module 114 for requesting access to the communication bus 102 therefrom (Fig. 5).

Packets received from the LAN segment 310 by the transceiver 308 are directed to the communication bus 102 through a receive FIFO buffer 316, while packets to be transmitted over the LAN segment 310 are directed from the communication bus 102 to the transceiver 308 through a transmit FIFO buffer 318. Preferably, the receive buffer 316 holds 128 bytes while the transmit buffer 318 holds 256 bytes. Note that an IEEE 802.3

data packet can include up to 1500 bytes of data in addition to the source address, the destination address and the frame check field. Thus, in the preferred embodiment, neither the receive buffer 316, nor the transmit buffer 318 is capable of storing a entire IEEE 802.3 data packet of the maximum size. An address latch 320 is also included in the port for latching addresses from the communication bus 102 and providing them to the transceiver 308.

Packet flow through the multi-port bridge 100 occurs in the following manner. A data packet, such as an IEEE 802.3 data packet, originating from a node (source node) in a segment of the local area network is received by a corresponding one of the ports 104-112 (source port) of the multi-port bridge 100 (Fig. 5). The receive buffer 316 in the source port receives the data packet as the packet is being received by the transceiver 308 in the source port from the LAN segment associated with the source port. After the first twelve bytes, corresponding to the source address and the destination address for the packet, are received, the receive finite state machine 312 requests a look-up cycle from the bus control circuit 114 (Fig. 5) by raising an interrupt request line coupled to the bus control module 114. The bus control module 114 monitors such requests and grants each request according to an appropriate priority. Upon granting the request, the bus control module 114 notifies the source port by placing a bit pattern identifying a bus grant on the command lines of the communication bus 102 and a bit pattern uniquely identifying the source port on the data lines of the communication bus 102.

The first four bytes of the destination address for the packet are then placed from the receive buffer 316 of the source port onto the data lines of the communication bus 102, while a corresponding bit pattern is placed on the command lines of the communication bus 102 by the bus controller 302 of the source port. The look-up control module 120 (Fig. 5) receives the first four bytes of the destination address. Then, the source port places the last two bytes of the destination address for the packet and the first two bytes of the source address for the packet on the data lines of the communication bus 102 and places a corresponding bit pattern on the command lines of the communication bus 102. The look-up control module 120 receives these four bytes. Finally, the source port places the last four bytes of the source address for the packet on the data lines of the communication bus 102 and places a corresponding bit pattern on the command lines. The look-up control module 120 also receives these four bytes. Thus, the destination address

and source address are transferred over the communication bus 102 in segments that are each four bytes long as this corresponds to the width (32 bits) of the data lines of the communication bus 102. It will be apparent, however, that the communication bus 102 can have a different number of data lines, in which case, a different number of bytes can be transferred at a time.

Once the look-up control module 120 has received the destination address and the source address for the packet, the look-up control module 120 so notifies the memory control module 118 (Fig. 5). The memory control module 118 then updates the look-up tables 204 (Fig. 5) by ensuring that the source address for the packet is stored in the look-up tables 204 in association with the identification of the source port for the packet. This ensures that the look-up tables 204 accurately reflect any changes that may have occurred in the LAN (referred to as a learning phase). The information stored during the learning phase is utilized for directing subsequent packets. Once the learning phase is complete, the memory control 118 module utilizes the look-up tables 204 to determine which port (destination port) is associated with the destination address for the packet.

As a result of performing the look-up cycle, the memory control module 118 forms a bit pattern referred to as a "triplet". Fig. 7 illustrates the triplet which includes three fields: a first field 400 containing the identification of the source port, a second field 402 containing the identification of the destination port, and a third field 404 containing a starting address assigned to the incoming packet in the packet buffers 206 of the memory device 200. The first field 400 and the second field 402 are each preferably one byte long. while the third field 404 is preferably two bytes long. It will be apparent, however, that the ordering of the fields of the triplet and the size of each field can be altered. If the source port and the destination port identified by the triplet are the same, this indicates that the source and destination nodes are on the same segment of the LAN (intra-segment communication) and, therefore, the packet does not need to be bridged. In such case, no further action is taken relative to the packet (the packet is filtered).

Otherwise, the memory control module 118 places the triplet on the data lines of the communication bus 102 and places a bit patten indicating that an "initial triplet" is ready on the command lines. Each port monitors the communication bus 102. If the port identified as the destination port in the triplet is not currently busy transmitting or receiving another packet, the destination port will configure itself to receive the packet directly from

the source port (cut-through). Alternately, if the triplet buffer 306 in the port identified as the destination port in the triplet is nearly full, the bus controller 302 of the destination port applies a jam request signal to the command lines of the high bus 102. The source port will receive the jam request and, in response, will discard the incoming packet and also send a jam packet over its associated segment. The jam packet will cause the node (source node) which is the source of the packet to discontinue sending the packet and attempt to resend the packet after a waiting period.

As illustrated in Fig. 7, the triplets are preferably of a uniform size. Therefore, the exact number of triplets that can be accommodated by a triplet buffer 306 of a port can be determined from the amount of space available in the triplet buffer 306 of the port. Accordingly, unlike prior arrangements, extra space does need to be provided in the port to accommodate a data packet having an unknown length. According to the present invention, however, the jam request is preferably generated by a destination port for a packet when the triplet buffer 306 in the port has space available to store several triplets (e.g. ten triplets). This provides the destination port an ability to store triplets for packets which are in the process of being loaded into the packet buffers 206. The triplet buffer 306 in each port is preferably sized relative to the associated packet buffers 206 in the memory device 200 such that there is little or no possibility that the packet buffers 206 will become full before any triplet buffer 306 becomes full.

Once the triplet has been placed on the communication bus 102, the source port initiates a series of memory write cycles for loading the packet from the receive buffer 316 of the source port into the packet buffers 206 in the memory device 200 starting at the memory address identified by the third field of the triplet. Preferably, the packet buffers 206 include a space allocated to each port for storing packets to be transmitted by the port. Alternately, the space is allocated to each port for storing packets received by the port; it should be noted, however, that only one and not both types of allocated space need be provided. Packets are written into the space allocated to the port in the packet buffers 206 in a circular fashion; each new packet will overwrite portions of the oldest packet in the allocated space.

The packet is preferably loaded into the packet buffers 206 a predetermined offset from the assigned address. This provides a location for storing a header for the packet once the packet has been completely loaded into the packet buffers 206. For example, the

header can include an identification number assigned to the packet, the triplet for the packet and a receive status for the packet. The receive status indicates whether or not the entire packet has been successfully received and loaded into the packet buffer 206.

Multiple memory write cycles are generally needed to transfer the entire packet into the packet buffers 206 as the remaining portions of the packet will generally include more than thirty-two bits (the number of data lines in the communication bus 102). Writing of the packet in the packet buffers 206 preferably occurs as the remainder of the packet is still being received into the receive buffer 316 of the source port. For this reason, the receive buffer 316 for each port need not be capable of storing an entire data packet. In addition, if the destination port is configured for cut-through, the destination port will receive the packet into its transmit buffer 318 directly from the communication bus 102 simultaneously with the write cycles for loading of the packet into the packet buffers 206. During such a cut-through operation, the packet is received into a transmit buffer 318 of the destination port for immediate transmission to the LAN segment associated with the destination port.

Once the entire packet has been loaded into the packet buffers 206, the memory control module 118 again places the triplet on the data lines of the communication bus 102 and places a bit pattern on the command lines identifying this as the "final triplet." It should be noted that the initial triplet and for the final triplet are preferably identical, while the bit patterns placed on the command lines of the communication bus 102 for identifying each of the triplets are distinct. The destination port will then store the triplet in the triplet buffer 306 located within the destination port. Thus, the packet is queued for transmission by the destination port.

Then, when the destination port is no longer busy, the destination port will retrieve the packet from the packet buffers 206. This is accomplished by the destination port requesting access to the communication bus 102 from the bus control module 114. When the request is granted, the bus control module 114 places a bit pattern indicating a bus grant on the command lines of the communication bus 102 and a bit pattern identifying the port on the data lines. Once the destination port gains control of the communication bus 102, the destination port then initiates a series of read operations over the communication bus 102 by placing the starting address in the packet buffers 206 of the packet (from the third field of the triplet for the packet) on the data lines of the communication bus 102 and places a bit pattern identifying a memory read operation on the command lines. In

response, the memory control module 118 accesses the packet in the packet buffers 206. Preferably, the destination port checks the receive status for the packet. If the receive status indicates that the packet was not received successfully, the memory read operation is preferably halted and no further action is taken relative to the packet.

In addition, in the event that a cut-through operation was initiated, but was unsuccessful, the packet will need to be retransmitted by the destination port. For example, the cut-through operation may have been unsuccessful if a data collision occurred during its transmission over the segment associated with the destination port. In such case, the packet is retrieved from the packet buffers 206 as described above.

While the destination port is receiving the packet into its transmit buffer 316 from the packet buffers 206 or directly from the source port (as during cut-through), the destination port begins transmitting the packet to the LAN segment associated with the destination port under control of the transmit finite state machine 314. For this reason, the transmit buffer 318 for each port need not be capable of storing an entire data packet. The packet is then received from the LAN segment by the destination node for the packet.

It will be apparent that rather than utilizing the packet buffers 206 to store queued packets, the receive buffer 316 and/or transmit buffer 318 of each port can be enlarged to store the entirety of each queued packet. Such an embodiment is less preferred, however, because the performance is likely to be lower, especially if the total storage available for temporarily storing the data packets is reduced. In addition, the inclusion of large buffers in each of the ports complicates the integration of the multi-port bridge 100 into a single integrated circuit (with the memory device 200 being external to the integrated circuit).

Fig. 8 illustrates a block schematic diagram of the switch engine 150 of the multi-port bridge 100, the memory device 200 and an external processor 500 according to the present invention. The switch engine 150 includes the bus control block 114, the MPU port and mailbox 116, the memory control block 118 and the look-up control block 120 illustrated in Fig. 5. A memory bus 502 interconnects the switch engine 150 and the external processor 500 to the memory device 200. Preferably, access to the memory device 200 by the switch engine 150 and the external processor 500 is implemented by a multiplexor included as part of the memory bus 502 and which multiplexor is controlled by the switch engine 150. The multi-port bridge 100, including the communication bus 102 (Fig. 5), is preferably implemented as an integrated circuit mounted to a printed circuit

board 504. The memory device 200 and the external processor 500 are also mounted to the printed circuit board 504.

As described above in reference to Figs. 5-7, the bridging and filtering functions of the multi-port bridge are performed primarily by the switch engine 150 and the buffer memory 200. Because the switch engine 150 and ports 104-112 are preferably implemented as a number of finite state machines interconnected via a communication bus 102, the multi-port bridge 100 provides a high bandwidth capacity for directing data packets through the multi-port bridge 100. Thus, according to the present invention, the external processor 500 is provided to perform tasks in support of the functioning of the multi-port bridge 100. These functions include: providing a communication port for enabling the nodes of the LAN to communicate with nodes of a dissimilar LAN or a WAN and for enabling the nodes of the LAN to communicate with a file server for the LAN; providing parameters for initializing registers of the switch engine; collecting data from the LAN for performing network management functions; and providing services to the switch engine. The mailbox interface according to the present invention allows the external processor 500 to provide these functions without the need to dedicate a large number of pins of the integrated circuit package to such an interface.

Preferably, the external processor 500 is implemented as a reduced instruction set computer (RISC) to improve speed performance. The external processor 500 can have its own dedicated resources 506, such as memory for storing operating software for the external processor 500 and for use by the external processor 500 as a scratch pad. In addition, when the external processor 500 performs the functions of a file server for the LAN, the resources 506 can include a mass storage device for storing application programs and data files which is accessible by the external processor 500. Also, when the external processor 500 performs the function of providing an interconnection of the LAN to a dissimilar LAN or to a WAN, the resources 506 can include a communication device, such as a telephone modem, an integrated services digital network (ISDN) interface, a T1 media interface or a T3 media interface which is accessible by the external processor 500. In addition, multiple external processors 500 can be coupled to the memory bus 502. In such case, additional resources can be provided for such multiple external processors, such as one or more mass storage devices and/or one or more communication devices.

Fig. 9 illustrates a block schematic diagram of an apparatus according to the present invention for periodically updating the look-up tables 204. The look-up tables 204 include three sub-tables 204A-C and are preferably a portion of the memory device 200 (Fig. 5) configured to emulate a CAM device. It will be apparent, however, that the look-up tables 204 could be included in one or more memory devices distinct from the memory device 200. For example, the look-up tables 204 can be one or more CAM devices.

The look-up control module 120 preferably performs a hashing algorithm on each address before the address is stored in the look-up tables 204 along with a corresponding port identification. The hashing algorithm preferably removes the most significant bits of each address and retains the least significant bits unchanged, though other hashing algorithms could be used. Preferably, the sixteen least significant bits are retained such that there are 2^{16} (65,536) possible addresses after hashing. Accordingly, each of the sub-tables 204A-C preferably has 2^{16} (65,536) memory locations. It will be apparent, however, that the hashing algorithm can result in a different number of remaining bits and, thus, a different number of memory locations in each sub-table.

Fig. 10 illustrates a time chart according to the present invention for periodically updating the address and port data stored in each of the three look-up sub-tables #1-3 (204A-C). Assume that at time 0.0, the multi-port bridge 100 is first powered-up or reset such that there are initially no entries in any of the sub-tables #1-3 (204A-C) (e.g. a power-up or reset sequence sets each table entry to all zeros). During the first 2.5 minute time period, the source address included in each data packet appearing at any of the ports of the multi-port bridge 100 is hashed to 16 bits by removing the most significant 32 bits. An identification of the port (source port) which received the data packet is stored in a location of each of tables #1 (204A) and #2 (204B) designated by the hashed address.

Preferably, the 32 address bits which were removed during the hashing algorithm are also stored along with the port identification in the appropriate tables. These remaining 32 bits are used for verifying the complete address in the event that two or more nodes share hashed addresses (e.g. they have the same least significant 16 address bits). If an entry has already been made for a hashed source address in either table (based upon a packet originated by a first node), the remaining 32 source address bits of the packet are compared to the 32 bits stored in the memory location designated by the hashed source address. If the remaining 32 do not match, this indicates that the data packet originated

from a second node having the same least significant bits as the first node. In such case, a linked list is formed for the affected table(s) for storing the remaining 32 bits of the address of the second node and the port identification for the port to which the second node is connected. If the remaining 32 bits match, this indicates that only one node (the first node) having the hashed source address has transmitted a data packet. In such case, linked lists are not required.

Also during the first 2.5 minute time period, the destination address included in each data packet received by the multi-port bridge 100 is hashed to sixteen bits. The memory controller 106 uses the hashed destination address to look-up in table #1 (204A) the appropriate port to which the data packet is to be directed for transmission to the destination node.

If no data is stored in the location designated by the hashed destination address, this indicates that the data packet is intended for a node that has not yet transmitted either a data packet or a "keep alive" packet. In such case, the multi-port bridge 100 does not have sufficient information to appropriately direct the packet. Therefore, the packet is preferably directed to every port except the port that received the packet (the packet is broadcast).

If data is stored in the location designated by the hashed destination address, the remaining destination address bits are compared to the 32 address bits stored in the that location to determine if they match. If there is not a match of the remaining address bits, this indicates that two or more nodes share the same 16 least significant address bits. When this occurs during a learning phase, a linked list is set up to store the port identification for each of the nodes which share the 16 least significant address bits. If there is a match of the remaining address bits, the identification of the port is utilized to appropriately direct the data packet to that port for transmission to the destination node.

If the port indicated by the destination address is the same as the port that received the packet, this indicates that the source node and destination node for the packet are in the same LAN segment. In such case, the multi-port bridge 100 preferably does not direct the packet to the port because the destination node receives the packet directly from the source node.

Upon the expiration of the first 2.5 minutes, sub-table #3 (204C) is cleared of all entries. The step of clearing the sub-table #3 can be skipped, however, if no entries have been made in sub-table #3 since start-up. Then, during the second 2.5 minute time period,

data is stored in sub-tables #2 (204B) and #3 (204C) in the same manner that data was stored in sub-tables #1 (204A) and #2 (204B) during the first 2.5 minute time period. Also during the second 2.5 minute time period, table #2 (204B) is utilized for look-up cycles. Upon the expiration of the second 2.5 minutes, sub-table #1 (204A) is cleared of all entries. Then, during the third 2.5 minute time period, data is stored in sub-tables #1 (204A) and #3 (204C) while table #3 (204C) is utilized for look-up cycles. Upon the expiration of the third 2.5 minute time period, table #2 (204B) is cleared of all entries. During the fourth 2.5 minute time period, data is stored in sub-tables #1 (204A) and #2 (204B) while table #3 (204C) is utilized for look-up cycles. Upon expiration of the fourth 2.5 minute time period, sub-table #3 is cleared.

Note that the steps performed in the fourth time period are identical to the steps performed in the first time period (with the exception that it is not necessary to clear the sub-table #3 at the end of the first 2.5 minute time period). Thus, the process repeats by cycling through the steps of the first, second, and third time periods.

According to the invention, no entry utilized in any look-up cycle is more than 5 minutes old. However, the age of each entry is not individually tracked.

The present invention has been described in terms of specific embodiments incorporating details to facilitate the understanding of the principles of construction and operation of the invention. Such reference herein to specific embodiments and details thereof is not intended to limit the scope of the claims appended hereto. It will be apparent to those skilled in the art that modifications may be made in the embodiment chosen for illustration without departing from the spirit and scope of the invention. Specifically, it will be apparent to one of ordinary skill in the art that the device of the present invention could be implemented in several different ways and the apparatus disclosed above is only illustrative of the preferred embodiment of the invention and is in no way a limitation.

Claims:

What is claimed is:

1. A method of controlling the flow of data packets in a multi-port bridge having a plurality of ports interconnected to a memory device by a communication bus, the multi-port bridge for interconnecting a plurality of segments of a local area network, the method comprising steps of:

a. receiving a data packet into a receive buffer in a source port for the packet, the packet having a destination address, a source address and a data field and the packet being received from a segment associated with the source port;

b. looking up the destination address in a table for determining a destination port for the packet;

c. assigning a location in the memory device to the packet;

d. forming a triplet for the packet wherein the triplet includes a first field for identifying the source port, a second field for identifying the destination port, and a third field for identifying the location assigned to the packet in the memory device;

e. placing the triplet on the communication bus a first time;

f. loading the packet into the memory device at the location in the memory device assigned to the packet;

g. placing the triplet on the communication bus a second time;

h. storing the triplet in a triplet buffer in the destination port;

i. receiving the packet into a transmit buffer in the destination port from the location assigned to the packet in the memory device; and

j. transmitting the packet to a segment associated with the destination port for the packet.

2. The method according to claim 1 wherein the step of looking up the destination address is performed while the data field is being received by the source port.

3. The method according to claim 1 wherein the step of loading the packet into the memory device is performed while the data field is being received by the source port.

4. The method according to claim 1 wherein the step of transmitting the packet is performed while the data field is being received from the memory device by the destination port.

5. A multi-port bridge for interconnecting a plurality of segments of a local area network, the multi-port bridge comprising:

a. a communication bus;

b. a memory device coupled to the communication bus;

c. a source port coupled to the communication bus, wherein the source port receives a packet from a source node within a segment of the local area network associated with the source port, wherein the source port initiates a look-up cycle upon receiving an address of a destination node for the packet and wherein a product of the look-up cycle is a triplet having a first field for identifying the source port, a second field for identifying a destination port for the packet, and a third field for identifying a location assigned to the packet in the memory device; and

d. the destination port for transmitting the packet to the destination node within a segment of the local area network associated with the destination port wherein if the destination port is not busy, the destination port receives the packet directly from the source port after the triplet is placed on the communication bus a first time, and if the destination port does not receive the packet directly from the source port, the destination port receives the packet from the memory device after the triplet is placed on the communication bus a second time.

6. The apparatus according to claim 5 wherein packet is stored in the memory device between the first time the triplet is placed on the communication bus and the second time the triplet is placed on the communication bus.

7. The multi-port bridge according to claim 5 wherein the source port comprises a receive buffer for receiving the data packet.

8. The multi-port bridge according to claim 7 wherein the data packet has a maximum expected size and the receive buffer is smaller than the maximum expected size.

9. The multi-port bridge according to claim 5 wherein the destination port comprises a transmit buffer for transmitting the data packet.

10. The multi-port bridge according to claim 9 wherein the data packet has a maximum expected size and the transmit buffer is smaller than the maximum expected size.

11. A method of controlling flow of packets in a multi-port bridge having a plurality of ports interconnected to a memory device by a communication bus. the multi-port bridge for interconnecting a plurality of segments of a local area network, the method comprising steps of:

a. receiving a packet having a destination address from a segment associated with a source port for the packet into a receive buffer in the source port;

b. looking-up the destination address in a table for determining a destination port for the packet;

c. assigning a location in the memory device to the packet;

d. forming a triplet for the packet wherein the triplet comprises a first field for identifying the source port, a second field for identifying a destination port for the packet, and a third field for identifying a location assigned to the packet in the memory device;

e. placing the triplet on the communication bus a first time;

f. loading the packet into the memory device at the location assigned to the packet; and

g. determining whether the destination port is available to receive the packet directly from the source port and, if the destination port is available to receive the packet directly from the source port, performing a step of receiving the packet into a transmit buffer in the destination port simultaneously with loading the packet into the memory device, and if the destination port is not available to receive the packet directly from the

source port, receiving the packet into the destination port from the memory after the destination port is available to receive the packet.

12. The method according to claim 11 further comprising a step of placing the triplet on the communication bus a second time after the step of loading is completed.

13. The method according to claim 12 further comprising a step of storing the triplet in the destination port during the step of placing the triplet on the communication bus a second time.

14. The method according to claim 11 further comprising a step of transmitting the packet performed while the packet is being received into the transmit buffer of the destination port.

15. The method according to claim 11 wherein the packet also has a data field and wherein the step of looking-up the destination address is performed while the data field is being received by the source port.

16. The method according to claim 11 wherein the step of loading the packet into the memory device is performed while the data field is being received by the source port.

17. The method according to claim 11 further comprising a step of transmitting the packet performed while the packet is being received from the memory device by the destination port.

18. A method of periodically updating entries in a look-up table for appropriately directing data packets through a multi-port bridge having a plurality of ports, each port having a port identification, wherein each of the data packets is originated by a respective one of a plurality of nodes, each node having a node address and each node corresponding to a respective one of the plurality of ports, the method comprising steps of:

a. storing in a first memory store and in a second memory store the node address of each node which originates a data packet during a first period of time in association with the port identification of the corresponding port;

b. storing in the second memory store and in a third memory store the node address of each node which originates a data packet during a second period of time in association with the port identification of the corresponding port; and

c. storing in the first memory store and in the third memory store the node address of each node which originates a data packet during a third period of time in association with the port identification of the corresponding port.

19. The method according to claim 18 further comprising a step of utilizing the first memory store for directing data packets during the first period of time.

20. The method according to claim 19 further comprising a step of utilizing the second memory store for directing data packets during the second period of time.

21. The method according to claim 20 further comprising a step of utilizing the third memory store for directing data packets during the third period of time.

22. The method according to claim 19 wherein a product of the step of utilizing the first memory store for directing data packets is a triplet for each packet, each triplet having a first field containing the identification of the source port corresponding to the node which originated the packet, a second field containing an identification of a destination port which corresponds to a node that is an intended recipient for the packet, and a third field containing a starting address assigned to the packet in a packet buffer.

23. The method according to claim 18 further comprising a step of hashing each node address before storing the node address in each of the first, second and third memory stores.

24. The method according to claim 18 wherein the second period of time begins upon expiration of the first period of time.

25. The method according to claim 24 wherein the third period of time begins upon expiration of the second period of time.

26. The method according to claim 25 further comprising a step of clearing the first memory store upon expiration of the second period of time.

27. The method according to claim 26 further comprising a step of clearing the second memory store upon expiration of the third period of time.

28. The method according to claim 27 further comprising a step of clearing the third memory store upon expiration of the first period of time.

29. The method according to claim 28 further comprising steps of repeating steps a, b, c, and d, upon expiration of the third period of time.

30. An apparatus for periodically updating entries in a look-up table for appropriately directing data packets through a multi-port bridge having a plurality of ports, each port having a port identification, wherein each of the data packets is originated by a respective one of a plurality of nodes, each node having a node address and each node corresponding to a respective one of the plurality of ports, the apparatus comprising:

a. a first memory store for storing the node address of each node which originates a data packet during a first period of time in association with the port identification of the corresponding port and the first memory store for storing the node address of each node which originates a data packet during a third period of time in association with the port identification of the corresponding port, wherein a second period of time occurs between the first period of time and the third period of time;

b. a second memory store for storing the node address of each node which originates a data packet during the second period of time in association with

the port identification of the corresponding port and the second memory store for storing the node address of each node which originates a data packet during the first period of time in association with the port identification of the corresponding port; and

c. a third memory store for storing the node address of each node which originates a data packet during the third period of time in association with the port identification of the corresponding port and for storing the node address of each node which originates a data packet during the second period of time in association with the port identification of the corresponding port.

31. The apparatus according to claim 30 wherein each node address is hashed before being stored.

32. The apparatus according to claim 30 wherein the first, second and third memory stores are each a CAM device.

33. The apparatus according to claim 30 wherein the first, second and third memory stores are each a RAM device which emulates a CAM device.

34. The apparatus according to claim 30 wherein the second memory store is cleared upon expiration of the third period of time.

35. The apparatus according to claim 34 wherein the third memory store is cleared upon expiration of the first period of time.

36. The apparatus according to claim 30 wherein the first memory store is utilized for directing packets through the multi-port bridge during the first period of time.

37. The apparatus according to claim 36 wherein a triplet is formed for each packet, each triplet having a first field containing the identification of the port corresponding to the node which originated the packet, a second field containing an

identification of a port which corresponds to an intended recipient for the packet, and a third field containing a starting address assigned to the packet in a packet buffer.

38. The apparatus according to claim 36 further comprising a step of utilizing the second memory store for directing data packets during the second period of time.

39. The apparatus according to claim 38 further comprising a step of utilizing the third memory store for directing data packets during the third period of time.

40. An apparatus for periodically updating entries in a look-up table for appropriately directing data packets through a multi-port bridge having a plurality of ports, each port having a port identification, wherein each of the data packets is generated by a respective one of a plurality of nodes, each node having a node address and each node corresponding to a respective one of a plurality of ports of the multi-port bridge, the apparatus comprising:

a. a first memory store;

b. a second memory store;

c. a third memory store; and

d. a control logic circuit coupled to each of the first, second and third memory stores, the control logic circuit for storing in the first memory store and in the second memory store the node address of each node which originates a data packet during a first period of time in association with the port identification of the corresponding port and for storing in the second memory store and in the third memory store the node address of each node which originates a data packet during a second period of time in association with the port identification of the corresponding port and for storing in the first memory store and in the third memory store the node address of each node which originates a data packet during a third period of time in association with the port identification of the corresponding port.

41. The apparatus according to claim 40 wherein each data packet contains the address of the node which generated the data packet.

42. The apparatus according to claim 40 wherein each node address is hashed before being stored.

43. The apparatus according to claim 40 wherein the first, second and third memory stores are each a CAM device.

44. The apparatus according to claim 40 wherein the first, second and third memory stores are each a RAM device which emulates a CAM device.

45. The apparatus according to claim 40 wherein the second period of time begins upon expiration of the first period of time.

46. The apparatus according to claim 45 wherein the third period of time begins upon expiration of the second period of time.

47. The apparatus according to claim 46 wherein the second memory store is cleared upon expiration of the third period of time.

48. The apparatus according to claim 47 wherein the third memory store is cleared upon expiration of the first period of time.

49. The apparatus according to claim 40 wherein the first memory store is utilized for directing packets through the multi-port bridge during the first period of time.

50. The apparatus according to claim 49 wherein a triplet is formed for each packet, each triplet having a first field containing the identification of the port corresponding to the node which originated the packet, a second field containing an identification of a port which corresponds to an intended recipient for the packet, and a third field containing a starting address assigned to the packet in a packet buffer.

51. The apparatus according to claim 49 wherein the second memory store is utilized for directing packets through the multi-port bridge during the second period of time.

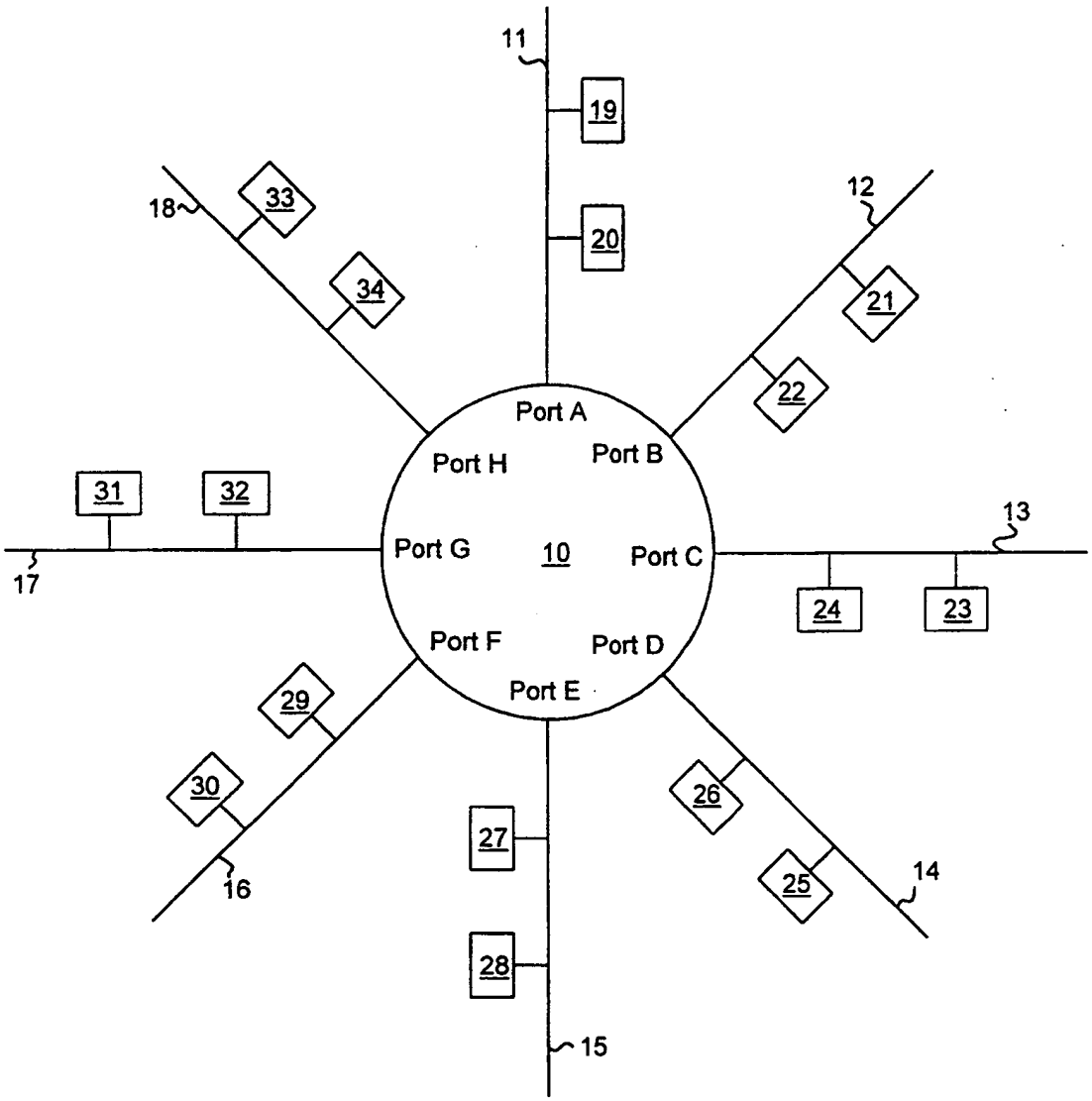


Fig. 1
(Prior Art)

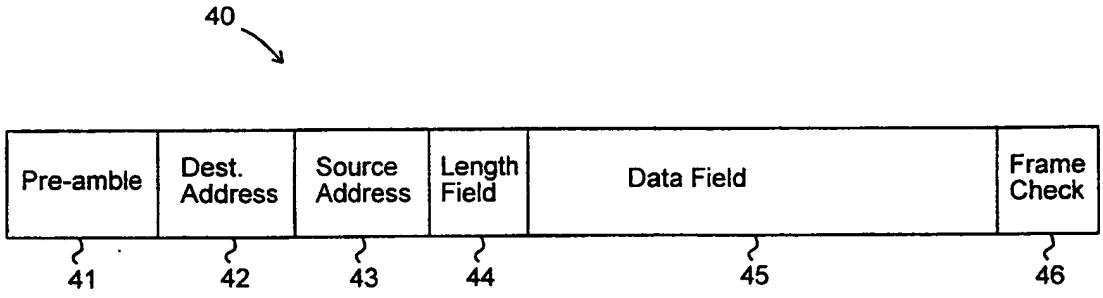


Fig. 2
(Prior Art)

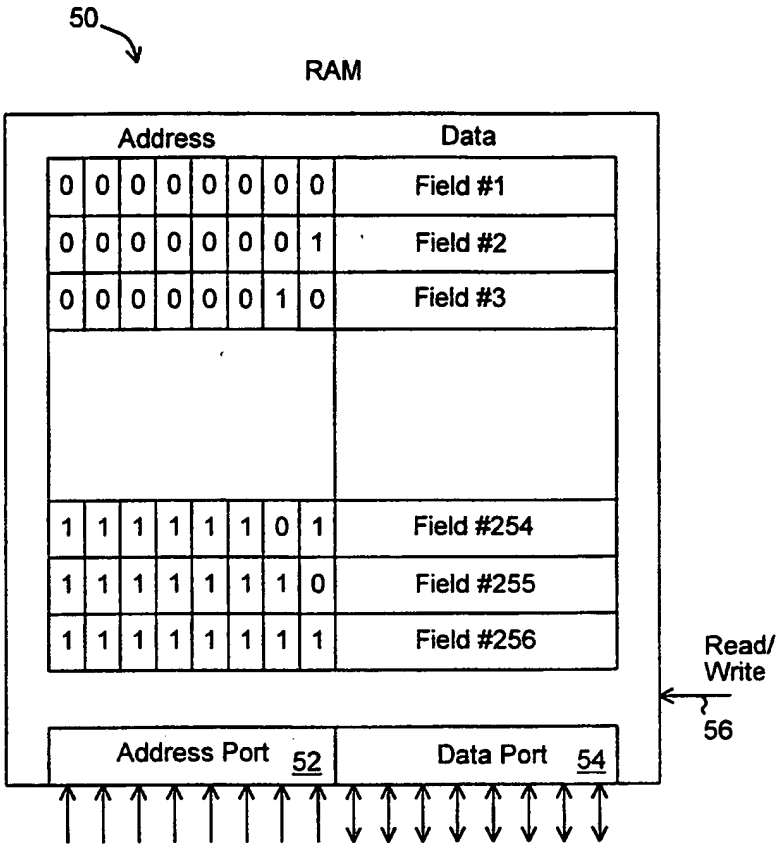


Fig. 3
(Prior Art)

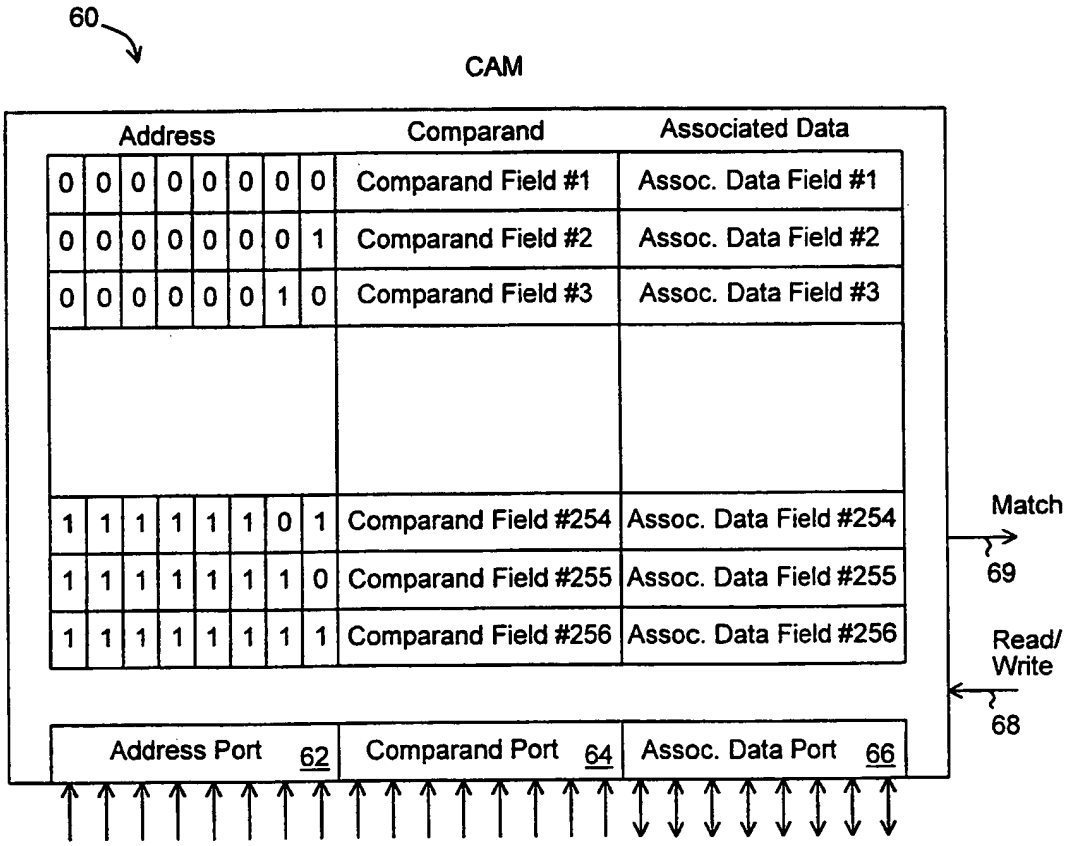


Fig. 4
(Prior Art)

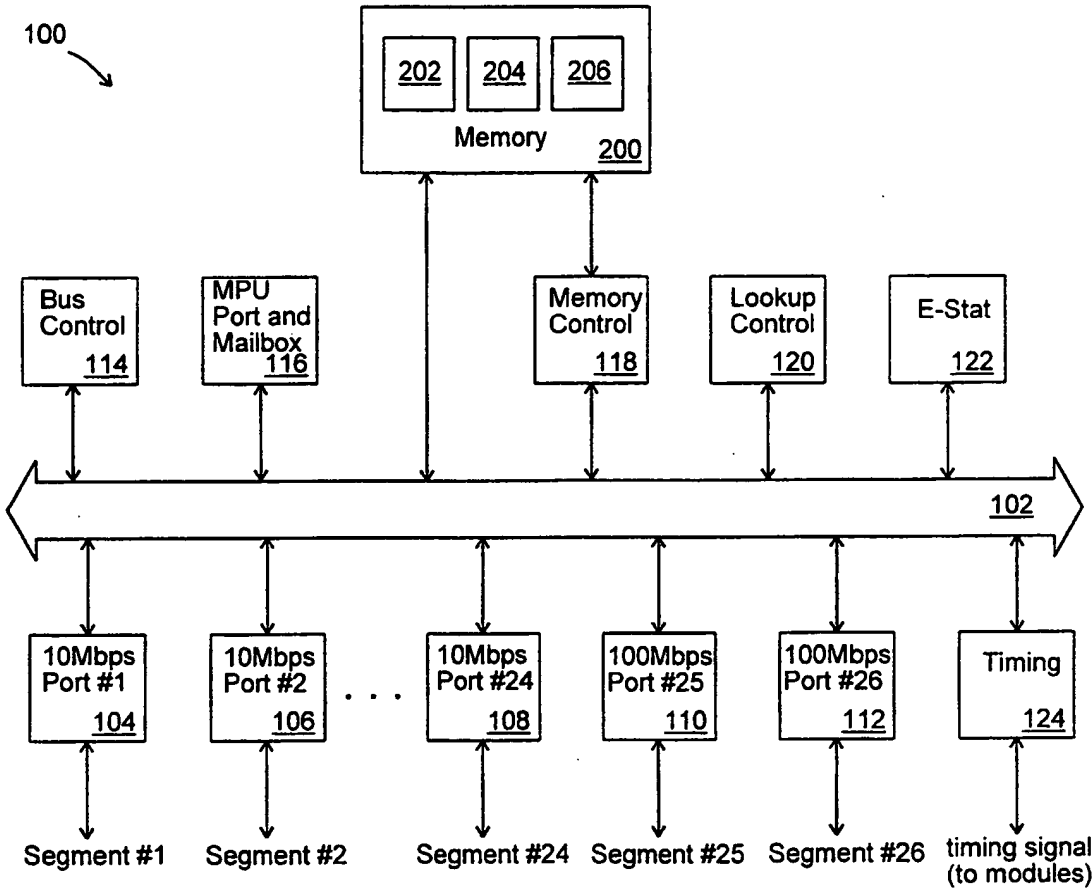

100
202
204
206
Memory
200
Bus Control
114
MPU Port and Mailbox
116
Memory Control
118
Lookup Control
120
E-Stat
122
102
10Mbps Port #1
104
10Mbps Port #2
106
10Mbps Port #24
108
100Mbps Port #25
110
100Mbps Port #26
112
Timing
124
Segment #1
Segment #2
Segment #24
Segment #25
Segment #26
timing signals (to modules)

Fig. 5

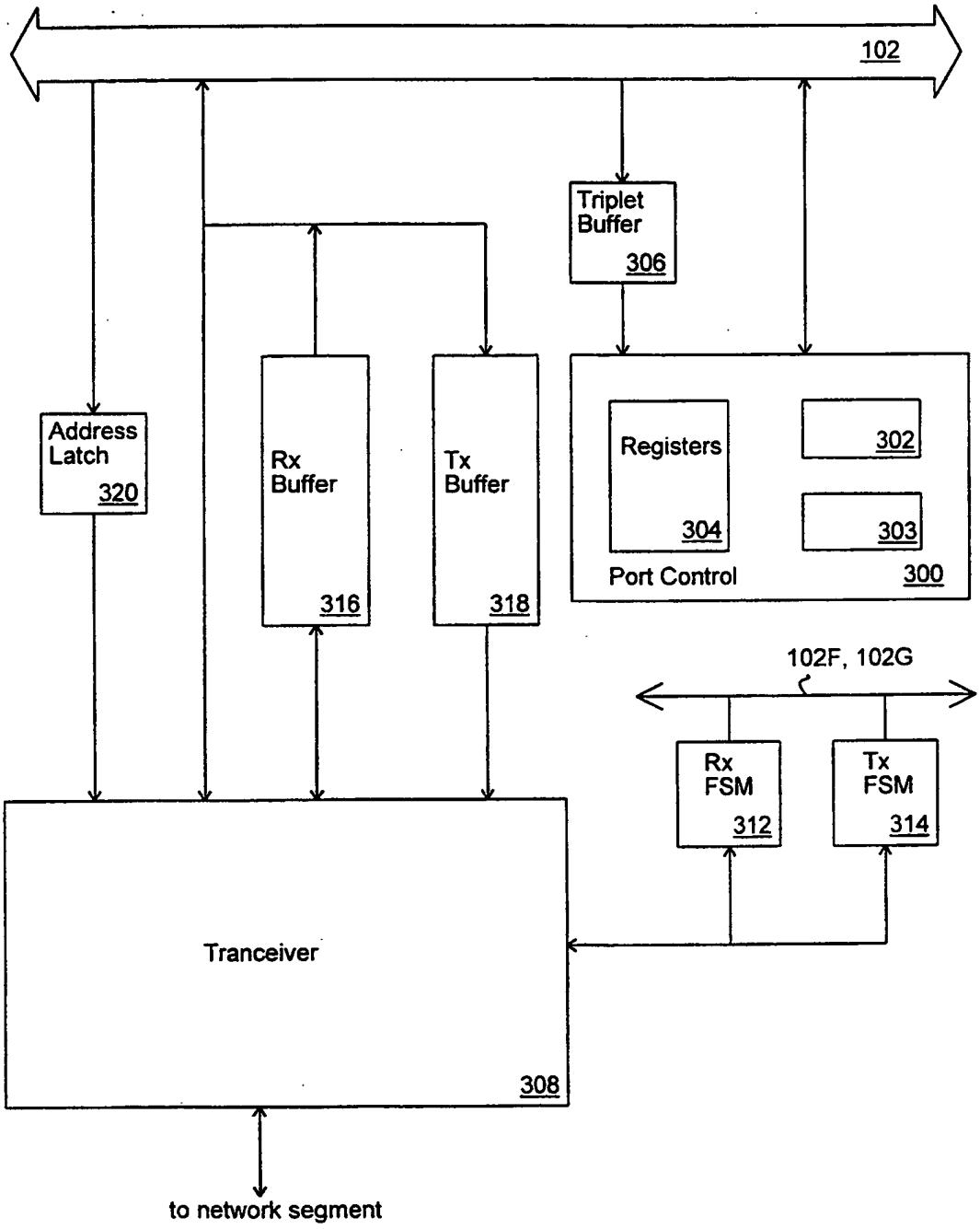

102
Triplet Buffer
306
Address Latch
320
Rx Buffer
316
Tx Buffer
318
Registers
304
302
303
Port Control
300
102F, 102G
Rx FSM
312
Tx FSM
314
Tranceiver
308
to network segment

Fig. 6

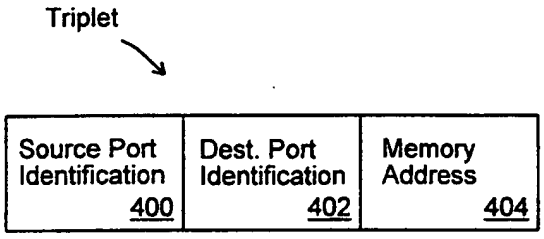

Triplet
Source Port Identification 400
Dest. Port Identification 402
Memory Address 404

Fig. 7

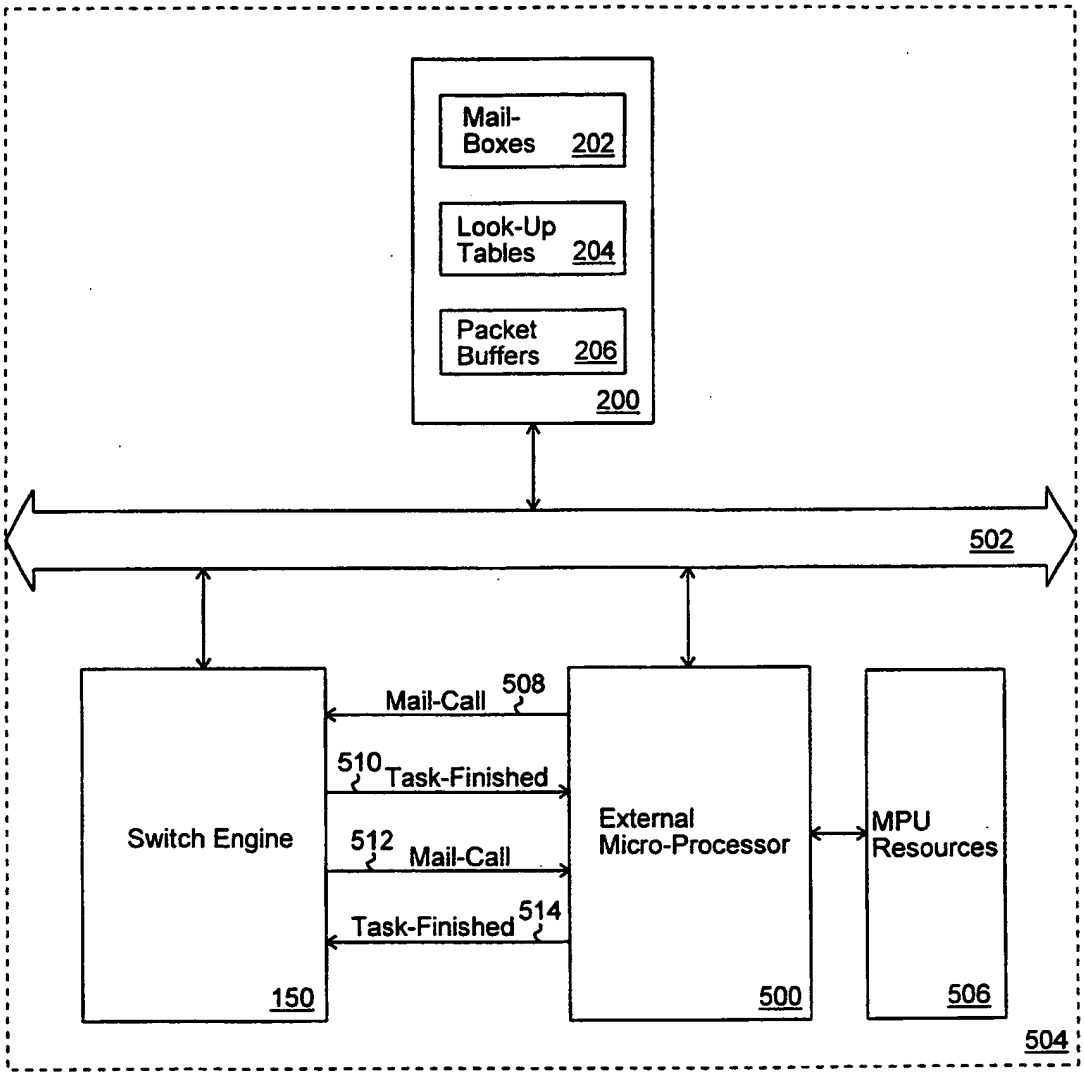

Mail-Boxes 202
Look-Up Tables 204
Packet Buffers 206
200
502
Mail-Call 508
510 Task-Finished
512 Mail-Call
Task-Finished 514
Switch Engine
150
External Micro-Processor
500
MPU Resources
506
504

Fig. 8

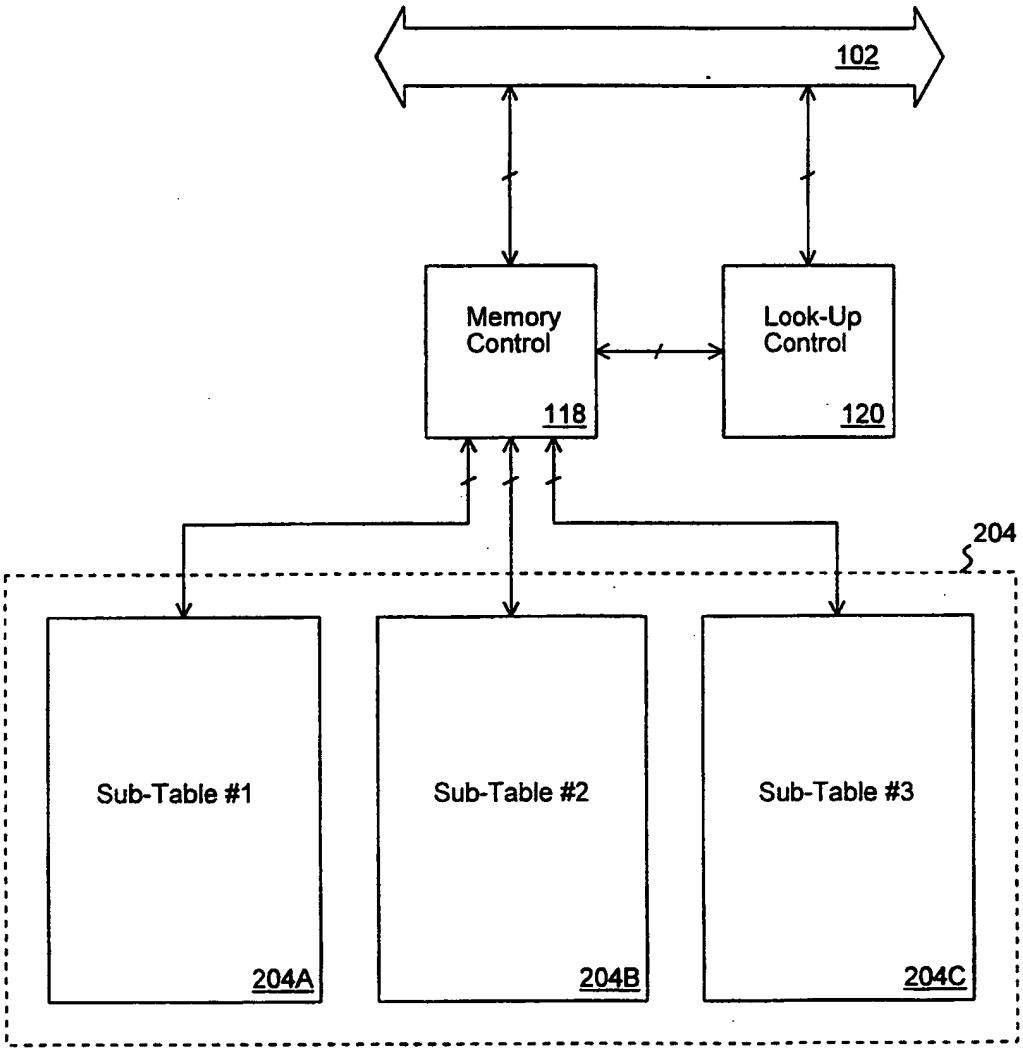

102
Memory Control
118
Look-Up Control
120
204
Sub-Table #1
204A
Sub-Table #2
204B
Sub-Table #3
204C

Fig. 9

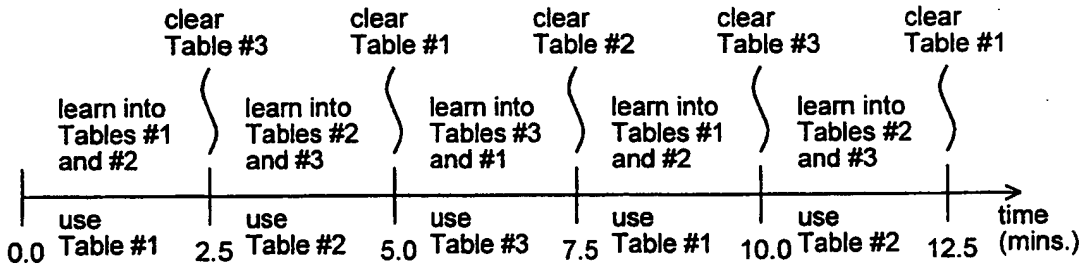

clear Table #3
clear Table #1
clear Table #2
clear Table #3
clear Table #1
learn into Tables #1 and #2
learn into Tables #2 and #3
learn into Tables #3 and #1
learn into Tables #1 and #2
learn into Tables #2 and #3
use Table #1
use Table #2
use Table #3
use Table #1
use Table #2
0.0
2.5
5.0
7.5
10.0
12.5
time (mins.)

Fig. 10

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 12/44 — A3

(11) International Publication Number: WO 99/14893

(43) International Publication Date: 25 March 1999 (25.03.99)

(21) International Application Number: PCT/US98/19169

(22) International Filing Date: 15 September 1998 (15.09.98)

(30) Priority Data:

60/059,171	17 September 1997 (17.09.97)	US
08/947,081	8 October 1997 (08.10.97)	US
09/050,750	30 March 1998 (30.03.98)	US

(71) Applicant *(for all designated States except US)*: SONY ELECTRONICS INC. [US/US]; 1 Sony Drive, Park Ridge, NJ 07656 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: CHUNG, David, H. [US/US]; Apartment 1901, 243 Buena Vista Avenue, Sunnyvale, CA 94086 (US).

(74) Agents: HAVERSTOCK, Thomas, B. et al.; Haverstock & Owens LLP, Suite 420, 260 Sheridan Avenue, Palo Alto, CA 94306 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, HR, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

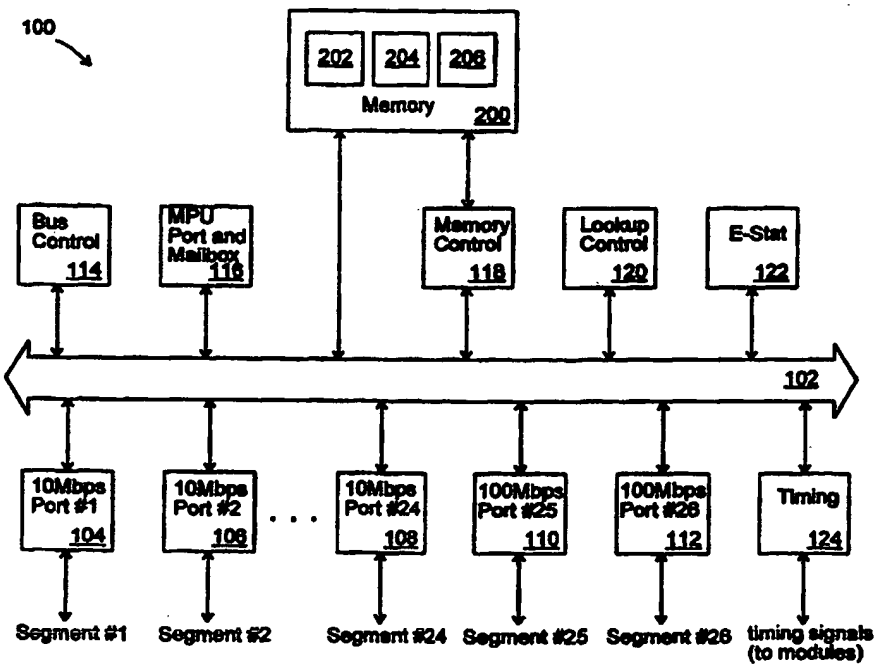
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(88) Date of publication of the international search report: 29 July 1999 (29.07.99)

(54) Title: MULTI-PORT BRIDGE WITH TRIPLET ARCHITECTURE AND PERIODICAL UPDATE OF ADDRESS LOOK-UP TABLE

(57) Abstract

A multi-port bridge includes a memory and a plurality of ports. Each port includes a receive buffer, a transmit buffer and a "triplet" buffer. As a data packet is being received by the receive buffer of a port, a look-up table is utilized to identify the appropriate destination port for the packet. A result of the look-up is a "triplet" which includes: a first field containing the identification of the source port, a second field containing the identification of the destination port, and a third field containing a starting address assigned to the incoming packet in the memory. The triplet is placed upon the communication bus a first time. If the destination port is available, the destination port receives the packet simultaneously as the packet is stored in the memory. Otherwise, the triplet is placed on the communication bus a second time after the packet is stored in the memory. The destination port stores the triplet in its triplet buffer. Then, when the destination port is available, the destination port retrieves the packet from the memory for transmission.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US 98/19169

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L12/44

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 597 789 A (IBM) 18 May 1994 see abstract see page 4, line 45 - page 5, line 9	1-17
A	KANAKIA H ET AL: "THE VMP NETWORK ADAPTER BOARD (NAB): HIGH-PERFORMANCE NETWORK COMMUNICATION FOR MULTIPROCESSORS" COMPUTER COMMUNICATIONS REVIEW, vol. 18, no. 4, 1988, pages 175-187, XP002035238 Section 2	1-17

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
15 February 1999	15. 06. 99

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	PEREZ PEREZ J. C.

2

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Im...ational application No.
PCT/US 98/19169

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This International Search Report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the International Application that do not comply with the prescribed requirements to such an extent that no meaningful International Search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

see additional sheet

1. ☐ As all required additional search fees were timely paid by the applicant, this International Search Report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this International Search Report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this International Search Report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

1-17

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 210

1. Claims: 1-17

A method of controlling the flow of packets in a multi-port bridge.

2. Claims: 18-51

A method for periodically updating entries in a look-up table.

INTERNATIONAL SEARCH REPORT

Information on patent family members

Internal Application No
PCT/US 98/19169

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0597789 A	18-05-1994	US 5448565 A	05-09-1995
		CA 2100543 A,C	13-05-1994
		JP 2537017 B	25-09-1996
		JP 6216913 A	05-08-1994